1. Обґрунтування вибору методу

Оскільки система є несиметричною, доцільно реалізовувати метод Гаусса.

2. Лістинг

```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;

int mul_func(float **mat1, float **mat2, float **res_mat, int m){

        int i, j, k;

        for(i = 0; i < m; i++){
                for(j = 0; j < m; j++){
                        res_mat[i][j] = 0;
                        for(k = 0; k < m; k++){
                                res_mat[i][j]=res_mat[i][j]+mat1[i][k]*mat2[k][j];
                        }
                }
        }

        return 0;
}

int gau_fun(float **mat, float *res, float &det, int m) {

        int i, j, k, max_index;
    float maxie;
    float matr[m][m + 1];

    for(i = 0; i < m; i++)
        for(j = 0; j < m + 1; j++)
            matr[i][j] = mat[i][j];
```

```
   for(k = 0; k < m; k++) {

              maxie = matr[k][k];

          for(i = k; i < m; i++) {
          if (matr[i][k] > maxie) {maxie = matr[i][k]; max_index = i;}
          }

          if(maxie == 0) return 0;

          for(int l = k + 1; l < m; l++) {
                 float koef = matr[l][k]/matr[k][k];
             for(j = 0; j < m + 1; j++)
                    matr[l][j] = matr[l][j] - matr[k][j]*koef;
      }
   }

   res[m - 1] = matr[m - 1][m]/matr[m - 1][m - 1];

       for(i = m - 2; i >= 0; i--) {
          res[i] = matr[i][m]/matr[i][i];
          for(j = i + 1; j < m; j++) res[i] = res[i] - matr[i][j]/matr[i][i]*res[j];
       }

       det = 1;
       for(i = 0; i < m; i++)
          for(j = i; j < i + 1; j++)
             det = det*matr[i][j];

       return 1;
}

int main() {

       int i, j, k, l, n;
       float deter;

       cout << "Input number of unknown variables: ";
       do {
              k = scanf("%i", &n);
              fflush(stdin);
              if (k != 1) cout << "Error, input value again: ";
       } while(k != 1);

       float **matrix = new float*[n];
       for(i = 0; i < n; i++)
          matrix[i] = new float[n + 1];

       float *result = new float[n];

       float **matrix_reverse = new float*[n];
       for(i = 0; i < n; i++)
```

```
    matrix_reverse[i] = new float[n];

  float **matrix_mult = new float*[n];
  for(i = 0; i < n; i++)
    matrix_mult[i] = new float[n];

  cout << "\nInput matrix A: \n";

  for(i = 0; i < n; i++)
    for(j = 0; j < n + 1; j++){
      cout << "Input a[" << i+1 << ", " << j+1 << "]: ";
              do {
             k = scanf("%f", &matrix[i][j]);
             fflush(stdin);
             if (k != 1) cout << "Error, input value again: ";
  } while(k != 1);
    }

  cout << "\nMatrix A: \n";

  for(i = 0; i < n; i++){
    for(j = 0; j < n + 1; j++)
      cout << matrix[i][j] << " ";
    cout << "\n";
}

  if(gau_fun(matrix, result, deter, n) == 0) {cout << "\nError!\n";}
  else {
     cout << "\nRoots obtained: \n";
   for(i = 0; i < n; i++){
       cout << result[i] << " ";
   }

   cout << "\n\ndet A = " << deter << "\n";

     for(k = 0; k < n; k++){
     matrix[k][n] = 1;
     for(l = 0; l < n; l++){
      if(l != k) matrix[l][n] = 0;
             }

             gau_fun(matrix, result, deter, n);

             for(i = 0; i < n; i++)
                matrix_reverse[i][k] = result[i];
      }

      cout << "\nReverse matrix: \n";

      for(i = 0; i < n; i++){
        for(j = 0; j < n; j++)
           cout << matrix_reverse[i][j] << " ";
        cout << "\n";
   }
```

```
        mul_func(matrix, matrix_reverse, matrix_mult, n);

            cout << "\nA*A^(-1): \n";

            cout.setf(ios_base::fixed);
        cout.precision(10);
            for(i = 0; i < n; i++){
                for(j = 0; j < n; j++)
                    cout << matrix_mult[i][j] << " ";
                cout << "\n";
        }


        cout <<"\n";
        float *b_res = new float[n];
            for (i=0; i<n; i++)
                        b_res[i]=matrix[i][0]*result[0] +
                        matrix[i][1]*result[1]+ matrix[i][2]*result[2]+
                        matrix[i][3]*result[3]+ matrix[i][4]*result[4];

            for(i=0; i<n; i++)
               cout <<"\n"<<matrix[i][n]-b_res[i];
    }

        return 0;
}
```

3. Результати роботи програми